

AF-# / Jhu

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re patent application of:

) Date: February 15, 2005

James S. Bratsanos, et al.

) Attorney Docket No.: E-989

Serial No.: 09/541,001

) Customer No.: 00919

Filed: March 31, 2000

) Group Art Unit: 2624

Confirmation No.: 1962

) Examiner: Thierry L. Pham

Title: **METHOD AND SYSTEM FOR MODIFYING PRINT STREAM DATA  
TO ALLOW PRINTING OVER A SINGLE I/O PORT**

**TRANSMITTAL OF APPEAL BRIEF (PATENT APPLICATION 37 CFR 1.192)**

Mail Stop Appeal Brief-Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Transmitted herewith in **triplicate** is the **APPEAL BRIEF** in the above-identified patent application with respect to the Notice of Appeal filed on December 30, 2004.

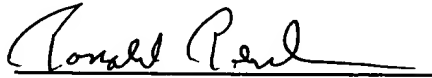
Pursuant to 37 CFR 1.17(c), the fee for filing the Appeal Brief is \$500.00.

Please charge Deposit Account No. **16-1885** in the amount of \$500.00 to cover the above fees.

The Commissioner is hereby authorized to charge any additional fees which may be required to Deposit Account No. **16-1885**.

A duplicate copy of this transmittal is enclosed for use in charging the Deposit Account.

Respectfully submitted,



Ronald Reichman  
Reg. No. 26,796  
Attorney of Record  
Telephone (203) 924-3854

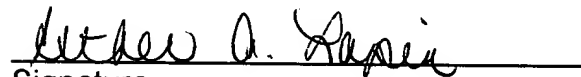
PITNEY BOWES INC.  
Intellectual Property and  
Technology Law Department  
35 Waterview Drive  
P.O. Box 3000  
Shelton, CT 06484-8000

**CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:

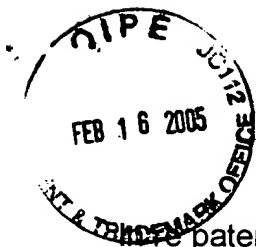
Mail Stop Appeal Brief-Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

On February 14, 2005  
Date of Deposit

  
Signature

Esther A. Lapin  
Name of Rep.

February 14, 2005  
Date



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

patent application of: ) Date: February 14, 2005  
James S. Bratsanos, et al ) Attorney Docket No.: E-989  
Serial No.: 09/541,001 ) Customer No.: 00919  
Filed: March 31, 2000 ) Group Art Unit: 2624  
Confirmation No.: 1962 ) Examiner: Thierry L. Pham

**TITLE: A METHOD AND SYSTEM FOR MODIFYING PRINT STREAM DATA TO  
ALLOW PRINTING OVER A SINGLE I/O PORT**

**APPELLANTS' BRIEF**

Mail Stop Appeal Brief – Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

This Brief is in furtherance of the Notice of Appeal filed in this case on December  
30, 2004.

This Brief is transmitted in triplicate.

**CERTIFICATE OF MAILING:**

I hereby certify that this correspondence is being deposited with the United States Postal Service  
as first class mail in an envelope addressed to:

Mail Stop Appeal Brief – Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

On February 14, 2005  
Date of Deposit

Esther A. Lapin  
Signature

Esther A. Lapin  
Name of Rep.

February 14, 2005  
Date

02/17/2005 MAHMED1 00000010 161885 09541001  
01 FC:1402 500.00 DA

**TABLE OF CONTENTS**

This Brief contains these items under the following headings and in the order set forth below:

- I. REAL PARTY IN INTEREST
- II. RELATED APPEALS AND INTERFERENCES
- III. STATUS OF CLAIMS
- IV. STATUS OF AMENDMENTS
- V. SUMMARY OF INVENTION
- VI. ISSUES
- VII. GROUPING OF CLAIMS
- VIII. ARGUMENT
- IX. PRAYER FOR RELIEF
- APPENDIX A CLAIMS IN THE APPEAL

**I. REAL PARTY IN INTEREST**

Pitney Bowes Inc. is the real party in interest by way of assignment from the Appellant.

**II. RELATED APPEALS AND INTERFERENCES**

There are no related Appeals or Interferences.

**III. STATUS OF CLAIMS**

- A. Claims 1-10 are in the application.
- B. Claims 1-10 are rejected.
- C. Claims 1-10 are on appeal.

**IV. STATUS OF AMENDMENTS**

An Amendment subsequent to the October 4, 2004, Final Office Action was filed on November 24, 2004. This Amendment was not entered.

**V. SUMMARY OF INVENTION**

**A. BACKGROUND**

**The prior art did not disclose or anticipate a method of modifying print stream data in a printing system so that one driver may be used to cause a document printer and an envelope printer to print from the same print stream.**

Mail preparation systems, such as the DOCUMATCH™ mail processing and finishing system available from Pitney Bowes Inc. of Stamford, Connecticut, establish a mail piece print run at a host personal computer (PC) and then direct the stream to printer peripherals for printing to an envelope and/or to a page substrate. Mail preparation systems are an example of systems whose purpose is to utilize address

lists, perform addressing hygiene through the use of address correction techniques, assign barcoding and, download data to addressing printers, collators, sealers, and the like, for the purpose of producing a mailpiece.

These systems sometimes have only a single input/output (I/O) port interface between the PC and the document printer. Thus, the current mail preparation systems are generally constrained by their printer hardware architecture.

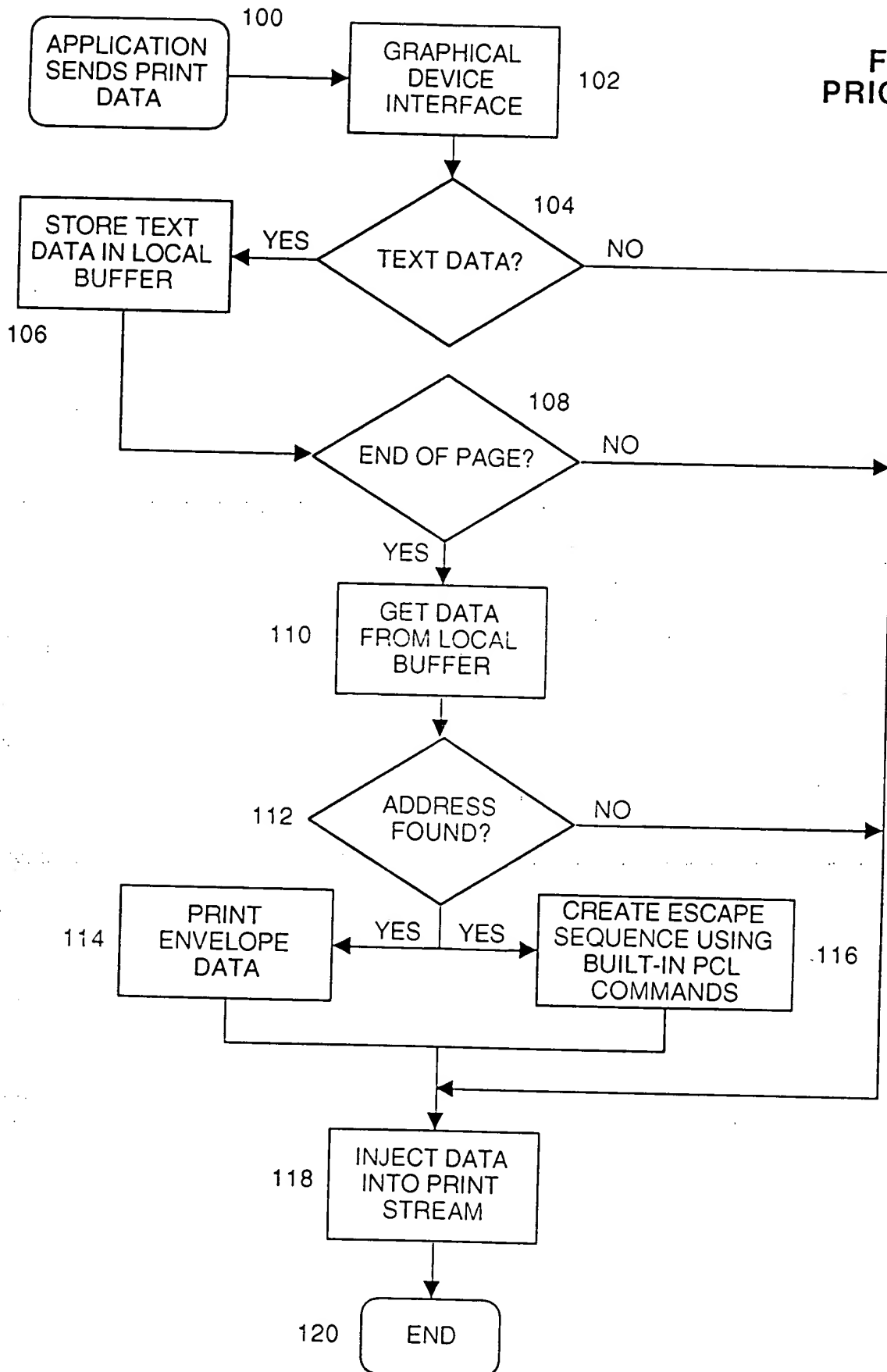
To support such system architecture, the application print data stream must be altered to allow the generation of envelope print data streams and its re-injection into the main application print stream. The creation of the envelope print stream involves the capture of text data contained in the document to generate the envelope, the use of an envelope definition module, and the use of proprietary print protocol language for the mail preparation system to direct the data to an appropriate printer.

The print stream created by the main application is generally in the form of text data, though it may take on other forms. The data must be parsed and checked before format correction and barcoding techniques can be directed to the addresses in the text for creation of a mailpiece.

Mailpiece production systems are known in the art and have developed with changes in postal service regulations (such as those of the United States Postal Service, or USPS) and with the proliferation of appropriate software applications. In turn, this production has served the need to automate and accelerate to accommodate growth.

A particular limitation to current methods and systems, however, is found in the assembly of the envelope print stream which fuels prior art systems. Mailpiece

FIG. 2  
PRIOR ART



production systems having two separate printers and a single I/O port are constrained by the serial connection between their printers, making it impossible for the document printer to query information directly from its corresponding envelope printer. Therefore, as illustrated by Appellants' Prior Art FIG. 2, which appears below.

FIG. 2, is a flowchart of the prior art method of printing envelope data extracted from a print stream in a WINDOWS 95 environment.

The prior art method begins at step **100** where a data processing application such as a mailpiece preparation application, operating in a WINDOWS 95 environment, initiates a print stream for each printed document. From step **100**, the method advances through a graphical device interface (GDI) at step **102** before entering a document driver module that begins at step **104**.

At step **104**, the method queries as to whether or not text data has been detected within the print stream. If the response to the query is "NO," then the method proceeds directly to step **118** where the data is re-injected into the print stream before advancing to step **120** where the sequence ends while the print stream is directed toward another peripheral device. If the response to the query at step **104** is "YES," however, then the method advances to step **106** where the text data is stored in a local buffer to await an "end-of-page" control mark from the system.

At step **108**, the method queries as to whether or not an end-of-page control mark has been received at the local buffer. If the response to the query is "NO," then the method proceeds directly to step **118** where the data is re-injected into the print stream before advancing to step **120** where the sequence ends while the print stream is directed toward another peripheral device. If the response to the query at step **108** is



"YES," however, then the method advances to step **110** where the data is retrieved from the local buffer before advancing to the query at step **112**.

At step **112**, the method queries as to whether an address has been found in the retrieved text. If the response to the query is "NO," then the method proceeds directly to step **118** where the data is re-injected into the print stream before advancing to step **120** where the sequence ends while the print stream is directed toward another peripheral device. If the response to the query at step **112** is "YES," however, then the method advances essentially simultaneously to steps **114** and **116**. At step **114**, the address data is printed to an envelope as envelope data, while at step **116**, an escape sequence is created using built-in printer command language (PCL) commands. Steps **114** and **116** rejoin at step **118** where the data is re-injected into the print stream before advancing to step **120** where the sequence ends while the print stream is directed toward another peripheral device such as a document printer.

**B. Appellants' claimed invention is a method of modifying print stream data in a printing system so that one driver may be used to cause a document printer and an envelope printer to print from the same print stream.**

1. Claim 1, the only independent claim in this patent application, relates to a method of modifying print stream data in a printing system. More particularly, claim 1 includes the following steps: determining, in a document driver, whether or not said print stream comprises text data, and: if said print stream comprises text data then tagging said text data and sending said tagged text data to a user mode module for further parsing; or if said print stream does not comprise text data then sending said print stream to a direct data injection step for a document printer; creating an envelope

printer device context from the document driver and transmitting said envelope data set to an envelope printer driver for creating an envelope printer device language file; and reading said printer device language and then injecting said envelope data set into said print stream so that the envelope data may be transmitted to the envelope printer and the document data to the document printer.

Appellants' invention is a method for modifying print stream data in a printing system having at least two printers and a single input/output port and comprises a number of steps and components.

The method begins by sending a print stream from a data processing application through a graphical device interface (GDI) to a print spooler to form a GDI print stream. The GDI print stream may contain: control data with a corresponding control page wizard which is utilized to facilitate mail merge functionality within the printing system; text data; address data; and/or, some other components. The data processing application can be a mailpiece designer application for preparing a mailpiece based on assigned parameters. Additionally, the mailpiece designer application is capable of presenting a data entry screen to a system user for performing the further steps of creating and/or modifying a mailpiece definition file and, storing and/or retrieving one or more mailpiece definition files wherein each of the files corresponds to a specific mail print run. In a preferred embodiment, the document designer application is a 32-bit WINDOWS automation server.

The printing system employs a print stream monitor within a document driver kernel context for scanning the GDI print stream to determine whether or not the print stream comprises a set of text data and/or a set of address data. If the print stream

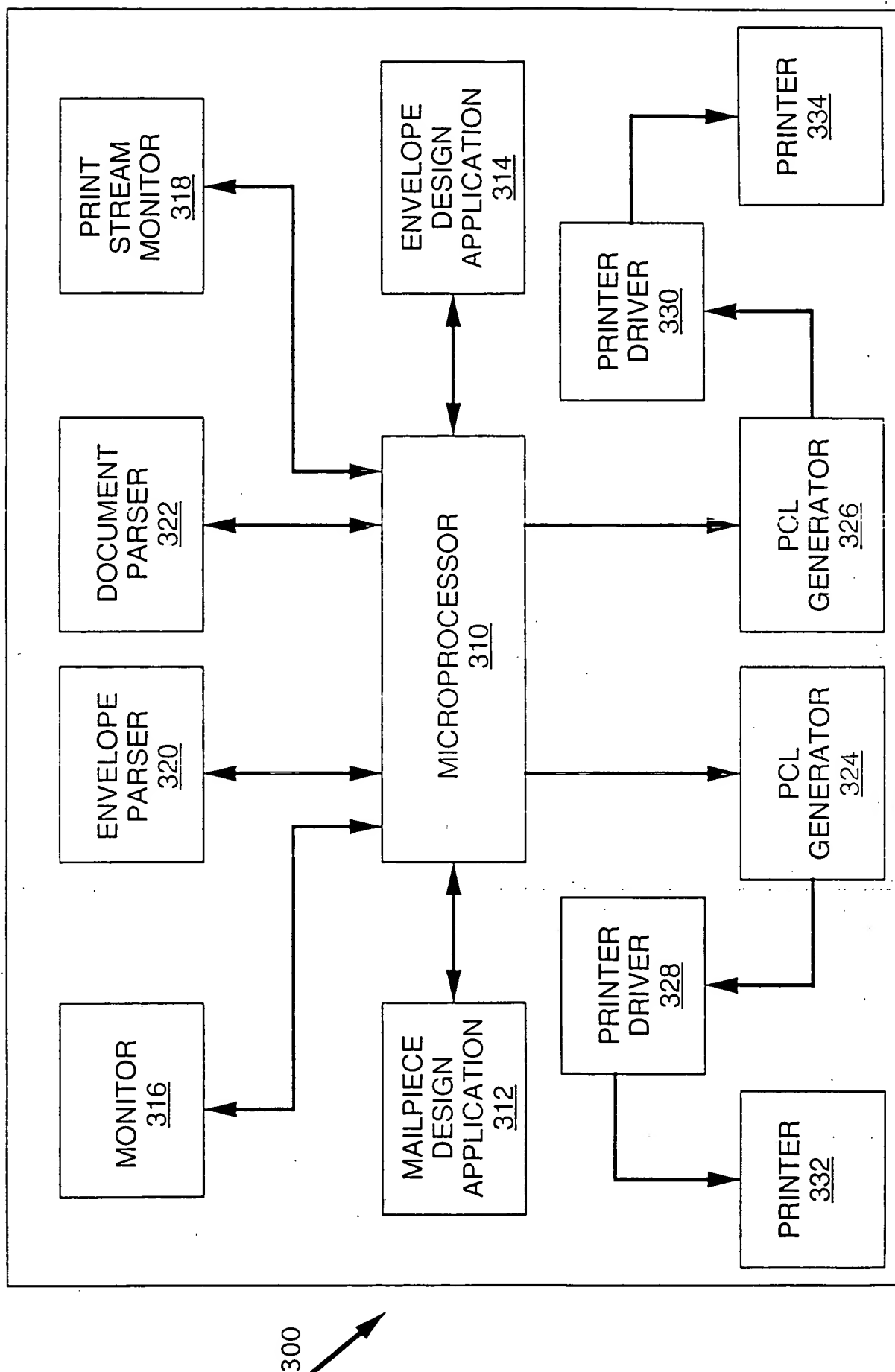
comprises text data then the text data is tagged and sent to a user mode module; however, if the print stream does not comprise text data, then the print stream is sent directly to a data injection step. After tagging, the text data is stored in a local buffer. The tagged text stored in the local buffer cannot be retrieved until the stored tagged text has received an end of page control mark for the text sought to be retrieved.

The tagged and stored text data is then retrieved from the local buffer and it is determined as to whether or not an address is contained within the tagged text. The determination is made by an envelope parser for detecting, parsing, and then extracting address data from the print stream. If an address is found in the tagged text, then the address is placed in an envelope print format to create an envelope data set; however, if an address is not found, then the tagged text is sent directly to the data injection step. An envelope printer device context is then created and the envelope data set is transmitted to an envelope kernel for creating an envelope printer device language file.

The GDI print stream is converted by a document printer command language (PCL) generator into an envelope printer language. The envelope data resulting is then utilized by a second designer application for displaying a set of data fields of the envelope data to a system user, reading a set of parameters created by the second designer application; and, writing the envelope data to a printer driver. The envelope data set is then printed.

Upon printing the envelope data set, the printer device language is then read by the print stream monitor which is used to modify the print stream by taking the envelope data set and injecting it back into the print stream from which it was extracted by merging the set of text data and the set of envelope data. The print stream is then

FIG. 4



transmitted to a next destination such as a document printer where a control page parser for detecting, parsing, and then extracting the document data from the print stream is employed. A printer command language (PCL) generator for converting the print stream into a document printer language is then employed. A printer driver is then activated for causing a printer to print the document data to one or more sheets.

Appellants' claimed invention is shown in Fig. 4 and described in line 3 of page 13 to line 3 of page 15 of Appellants' Patent Application. A copy of Fig. 4 appears next to this page.

**FIG. 4** is a block diagram of the system and corresponding components of the present invention.

A microprocessor **310** is shown interoperatively connected to a document design application **312**, for preparing a document such as a mailpiece with its associated text insert, to be printed as document text and as envelope text. In a preferred embodiment of the present invention, the document designer application **312** is a 32-bit WINDOWS automation server. The document designer application is capable of creating and/or modifying a mailpiece definition file and storing and/or retrieving one or more mailpiece definition files wherein each of said files corresponds to a specific mail print run and results in a print stream. Also connected to microprocessor **310**, is an envelope design application **314**.

The envelope design application **314** is utilized for displaying a set of data fields of the envelope text data portion to a system user; reading a set of parameters created by the envelope designer application; and, writing the envelope data to a printer driver. The set of data fields displayed is representative of the face of an envelope (comprising

an indicia print field and an addressee print field), thus allowing the system user a convenient way to check on field selection and placement.

Additionally connected to microprocessor **310** is a print stream monitor **318** for: scanning the print stream generated by the mailpiece design application **312**; detecting a set of document data or control data and a set of envelope data; interfacing with the envelope parser **320** to extract an address from the document text data; interfacing with the document parser **322** to extract control page information from the print stream; generating the envelope PCL print data at the PCL print generator **324**; and, for modifying the print stream to merge the two sets of data. The print stream monitor **318** maintains the system timing during printing of the mailpiece and the general performance of the document print job.

The system includes a document (or control page) parser **322** for detecting, parsing, and then extracting the document data from the print stream, as well as instructing the print stream task manager (not shown). Further included, is an envelope parser **320** for detecting, parsing, and then extracting the envelope data from the print stream and then indicating to the print stream task manager that an address has been detected.

To print the envelope, a PCL generator **324** is connected to the microprocessor **310** for converting the envelope data as extracted from the print stream into a second printer language, thus creating a proper PCL for the envelope text to be printed through printer driver **328** to printer **332** and on to an envelope or similar substrate. To print the document, a PCL generator **326** is connected to the microprocessor **310** for converting the document data as extracted from the print stream into a second printer language,

thus creating a proper PCL for the document text to be printed through printer driver **330** to printer **334** and on to one or more sheets or similar substrate. The WINDOWS NT printer architecture requires that every printer driver be implemented as a pair of user mode dynamic link libraries (DLL), as well as a printer specific component.

The WINDOWS NT printing architecture is part of the NT graphics architecture and consists of three components; these are: the applications that interface with the WINDOWS GDI; the server print spooler that interfaces with the print services convention; and, the kernel mode print services that include the printer driver minidriver and the I/O port interface. The print spooler internally accesses the user interface and the user mode printer driver components. The minidriver is a data file that contains the printer data tables as well as code specific to system driver that works in conjunction with the shim common driver. The purpose of the shim common driver is to intercept the raster graphic entry points to obtain print stream data and to provide each of the other drivers with common kernel functions when necessary.

## **VI. ISSUES**

Whether claims 1-10, are patentable under 35 USC §102(b) as being anticipated by Cordery, et al. (U.S. Patent No. 5,628,249).

## **VII. GROUPING OF CLAIMS**

Claims 1-10 are grouped together and stand and fall together.

## **VIII. ARGUMENT**

**Claims 1-10 have been rejected by the Examiner under 35 USC §102(b) as being anticipated by Cordery, et al. (U.S. Patent No. 5,628,249).**

Cordery discloses the following in line 53 of column 3 to line 6 of column 4:

"Driver **37** extracts the address from the document data in any convenient conventional manner, such as by the use of a predetermined field within the document data, or the use of an algorithm based upon the detection of alphanumeric combinations typical of zip codes, state names, city names, etc., as is also known, Driver **37** also accesses data store **38** to obtain the attribute information which includes processing attributes **40**, such as feeder selection, fold type, sealing mode etc. Preferably driver **37** also gets job type data **42** from data source **38** for inclusion in job header **12**. Driver **7** then adds separators **26-1** through **26-4** to create header **12** and records **14** as described above. As noted, generally each mail piece in a mailing job will be produced in an identical manner and the default values used for each mail piece. Accordingly, mail piece header **18** can be filled with null data or with copies of job header **12**. However, if it is desired to produce mailing jobs having mail pieces with varying attributes it would be well within the skill of a person of ordinary skill in the programming arts to modify a word processing application or produce a special application which would generate varying data for mail piece header **18**."

Cordery does not disclose or anticipate how one driver, namely driver **37**, may be used to cause a document printer and an envelope printer to print from the same print stream.

Cordery does not disclose or anticipate steps b(i), b(ii), e and f of claim 1, as amended, and those claims dependent thereon, namely, if said print stream comprises text data then tagging said text data and sending said tagged text data to a user mode module for further parsing; or if said print stream does not comprise text data then sending said print stream to a direct data injection step for a document printer; creating an envelope printer device context from the document driver and transmitting said envelope data set to an envelope printer driver for creating an envelope printer device language file; and reading said printer device language and then injecting said envelope



Serial No.: 09/541,001  
Attorney Docket No.: E-989  
Appellants' Brief: February 14, 2005

data set into said print stream so that the envelope data may be transmitted to the envelope printer and the document data to the document printer.

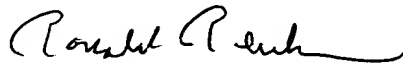
Typically a printer driver causes a printer to print envelopes and another printer driver causes a printer to print documents.

A document driver is used by Applicants to print to an envelope driver using a device context. The foregoing enables a single print stream to be transmitted to a document printer. This enables one to drive two printers with a single I/O link.

#### **IX. PRAYER FOR RELIEF**

Appellants respectfully submit that appealed claims 1-10 in this Application are patentable. It is requested that the Board of Appeal overrule the Examiner and direct allowance of the rejected claims.

Respectfully submitted,



---

Ronald Reichman  
Reg. No. 26,796  
Attorney of Record  
Telephone (203) 924-3854

PITNEY BOWES INC.  
Intellectual Property and  
Technology Law Department  
35 Waterview Drive  
P.O. Box 3000  
Shelton, CT 06484-8000

## **APPENDIX A – CLAIMS IN THE APPEAL**

1. A method of modifying print stream data in a printing system, said method comprising the steps of:

(a) sending a print stream from a data processing application to a print spooler;  
(b) determining, in a document driver, whether or not said print stream comprises text data, and:

(i) if said print stream comprises text data then tagging said text data and sending said tagged text data to a user mode module for further parsing; or

(ii) if said print stream does not comprise text data then sending said print stream to a direct data injection step for a document printer;

(c) storing said tagged text in a local buffer;

(d) retrieving said tagged text from said local buffer and determining whether or not an address is contained within said tagged text, and:

(i) if an address is found in said tagged text, then placing said address in an envelope print format to create an envelope data set; and

(ii) if an address is not found then sending said tagged text directly to said data injection step;

(e) creating an envelope printer device context from the document driver and transmitting said envelope data set to an envelope [kernel]printer driver for creating an envelope printer device language file;

- (f) reading said printer device language and then injecting said envelope data set into said print stream so that the envelope data may be transmitted to the envelope printer and the document data to the document printer; and
- (g) transmitting said print stream to a next destination.

2. The method of claim 1, wherein said print stream is passed through a graphical device interface (GDI) when being sent from said data processing application to said print spooler to form a GDI print stream.

3. The method of claim 1, wherein said print stream comprises control data.

4. The method of claim 1, wherein said local buffer stores said tagged text until at least one end-of-page control mark is received in said local buffer.

5. The method of claim 1, wherein said tagged text stored in said local buffer cannot be retrieved until said stored tagged text has received an end of page control mark for said stored tagged text sought to be retrieved.

6. The method of claim 1, wherein said data processing application is a mailpiece designer application.

7. The method of claim 6, wherein said mailpiece designer application is capable of presenting a data entry screen to a system user for performing the further steps of:

- (a) creating and/or modifying a mailpiece definition file; and
- (b) storing and/or retrieving one or more mailpiece definition files wherein each of said files corresponds to a specific mail print run.

8. The method of claim 1, wherein said print stream comprises a control page wizard.

9. The method of claim 8, wherein said control page wizard is utilized to facilitate mail merge functionality within said printing system.

10. The method of claim 2, wherein said GDI print stream is converted by a document printer command language (PCL) generator into an envelope printer language.